Machine learning and surrogate models for simulation-based optimization of accelerator design

Remi Lehe*[1], Adi Hanuka[2], Auralee Edelen[2], Xiaobiao Huang[6], Christopher Mayes[2], Nathan Cook[3], Axel Huebl[1], Claudio Emma[2], Blagoje Djordjević[9], Andreas Kemp[9], Scott Wilks[9], Ryan Roussel[4], Maxence Thévenet[5], Jean-Luc Vay[1], Ji Qiang[1], E. Paulo Alves[2,7], Chris Tennant[8], Andreas Adelmann[10], Henri Vincenti[11], Daniele Filippetto[1], Spencer Gessner[2], Alexander Scheinker[12], Alec Thomas[13], Daniel Winklehner[14], George K. Holt[15], Samuel R. Yoffe[15], Dino A. Jaroszynski[15], Nico Hoffmann[16], Alexander Debus[16], and Michael Bussmann[17,16]

[1]Lawrence Berkeley National Laboratory, Berkeley, CA, USA
[2]SLAC National Laboratory, Menlo Park, CA, USA
[3]RadiaSoft LLC, Boulder, CO, USA
[4]University of Chicago, Chicago, IL, USA
[5]DESY, Hamburg, Germany
[6]Argonne National Laboratory, Lemont, IL, USA
[7]University of California Los Angeles, Los Angeles, CA, USA
[8]Thomas Jefferson National Accelerator Facility, Newport News, VA, USA
[9]Lawrence Livermore National Laboratory, Livermore, CA, USA
[10]Paul Scherrer Institut, Villigen, Switzerland
[11]LIDYL, CEA-Université Paris-Saclay, CEA Saclay, 91 191 Gif-sur-Yvette, France
[12]Los Alamos National Laboratory, Los Alamos, NM, USA
[13]University of Michigan, Ann Arbor, MI, USA
[14]MIT, Cambridge, MA, USA
[15]University of Strathclyde, Glasgow, United Kingdom
[16]Helmholtz-Zentrum Dresden - Rossendorf (HZDR), Dresden, Germany
[17]Center for Advanced Systems Understanding (CASUS), Goerlitz, Germany

September 1, 2020

1 Simulation-based optimization of accelerator design

When designing new particle accelerators – or when designing new modes of operation for existing accelerators – it is essential to perform *design optimization*. This involves tuning a number of design parameters (e.g. strength of focusing elements, accelerating phase and accelerating gradient, positions of beamline elements, etc.) so as to maximize one or several measures of accelerator performance (e.g. beam quality, final energy, dynamic aperture, energy efficiency, etc.). The need for this type of optimization is ubiquitous across all types of accelerator experiments and facilities, including linacs, storage rings, free-electron lasers, and laser-plasma accelerators.

In general, evaluating the accelerator performance for *one set of design parameters* requires a sophisticated numerical simulation. Therefore, in order to search the high-dimensional space of design parameters and find the optimal point (or the Pareto front in the case of multi-objective optimization), many numerical simulations need to be run. However, advanced and realistic numerical simulations can be computationally expensive. It is therefore crucial, when searching the space of design parameters, to carefully choose the points that will be simulated so as to minimize the number of simulations needed to reach the optimal point (or the Pareto front).

2 Advantages of surrogate-model optimization

Historically, the accelerator community has used a variety of optimization algorithms to choose points to simulate, including the Nelder-Mead simplex algorithm, genetic algorithms, and particle swarm algorithms. However, while these optimization algorithms are robust and computationally cheap, they extract a limited amount of information from previously simulated points, and, as a result, they may still require many simulations to reach the optimal point or the Pareto front.

One alternative approach is to use a *surrogate model* in the optimization process. A surrogate model can *approximately* predict the performance of the accelerator for new sets of design parameters based

*rlehe@lbl.gov

on the knowledge extracted from previously simulated points. The surrogate model is much faster to execute than the actual numerical simulation. As such, a surrogate model can inform the choice of the next points to simulate, so as to select only the *most promising* points.

While a number of machine learning techniques could be used to build surrogate models, in practice recent research in the accelerator community has focused on two techniques: neural networks (NN) [1, 2, 3] and Gaussian processes (GP) [4, 5]. Using these techniques, it was shown that, for certain accelerator design problems, surrogate-model optimization requires significantly less simulations than traditional genetic algorithms – in some cases by one or two orders of magnitude [1, 5].

3 Needs and challenges in the near future

These early results illustrate the potential of surrogate-model optimization for accelerator design. Nevertheless, much remains to be done to address the needs of the community in the near future. Here, we suggest two areas of development.

3.1 Exploring best practices for surrogate-model optimization

There are many choices in the details of the surrogate optimization algorithms, and the community needs to build up knowledge and experience in finding the best practices that will suit a given accelerator design problem.

For instance, one key aspect is the sampling strategy for new simulated data points, as it affects both the accuracy of the surrogate model and the convergence speed of the optimization. The community needs to explore various aspects, such as how to balance exploration and exploitation, how often to retrain the surrogate model and with how much data, and how to sample the data in a parallel, asynchronous manner. For some of those issues, methods from the vast literature on Bayesian optimization and reinforcement learning need to be tested and applied in the context of accelerator design. There is also a need for a reliable estimate of the uncertainty on the *approximate* prediction of the surrogate model. This uncertainty appears naturally in the case of GP, but is dependent on the choice of hyperparameters. On the other hand, NN do not naturally feature an uncertainty, and thus extensions have to be considered (e.g. ensembles of NN, Bayesian NN, deep GPs).

Finally, it would also be desirable to have the capability to extract information from low-cost simulations (e.g. simulations at low resolution, with reduced geometry, or with simplifying assumptions) in order to guide the optimization of costly, more realistic simulations. In order to address this need, the community could take advantage of the literature on transfer learning and multi-fidelity Bayesian optimization.

3.2 Surrogate-model optimization framework

Surrogate-model optimization is a complex workflow that interleaves detailed numerical simulations, efficient sampling of new data points, and training of machine learning models. There is a strong need for a robust and flexible framework that enables users to rapidly setup this workflow. Ideally, this framework should be able to interface with high-performance computing facilities (through their batch submission system), support common machine learning techniques (NN and GP) while leaving enough flexibility to the user to customize these algorithms, and support multi-objective, multi-fidelity and asynchronous workflows.

Currently, there does not seem to be a single framework that simultaneously satisfies those requirements. Instead, different research groups have been building upon different existing open-source frameworks (e.g. libEnsemble [6, 7], xopt [8], Fireworks [9, 10], Ocelot [11]), in order to use or implement some of the above requirements. Given the diversity of accelerator applications and the exploratory state of the field, it is unclear at this point whether a single framework can be flexible enough to support all use cases. However, as the field matures, synergies between research groups and consolidation of their respective frameworks could be sought. One first step in this direction could be the creation of an open-source repository to share reproducible examples of surrogate-model optimization from various groups. More broadly, this effort will also be facilitated by the adoption of shared data standards and standardized modeling workflows across the community [12].

4 Conclusion

Surrogate-model optimization of accelerator designs is a nascent and promising area of research. By building upon recent development in machine learning tools, this field holds the potential to accelerate the optimization of particle accelerators, while making optimal use of available computational resources. In addition, thanks to the savings in the number of full scale simulations that are needed with these methods, each simulation can afford to be more precise (e.g., higher resolution, higher-order models, better description of halo) and/or complete (e.g., more physics, larger fraction of the machine up to start-to-end modeling, virtual accelerators [13]). This will foster breakthroughs in beam and accelerator physics, enabling the design, development and operation of accelerators that would not even be possible otherwise.

References

[1] Auralee Edelen, Nicole Neveu, Matthias Frey, Yannick Huber, Christopher Mayes, and Andreas Adelmann. Machine learning for orders of magnitude speedup in multiobjective optimization of particle accelerator systems. *Physical Review Accelerators and Beams*, 23(4), Apr 2020.

[2] Faya Wang, Minghao Song, Auralee Edelen, and Xiaobiao Huang. Machine learning for design optimization of storage ring nonlinear dynamics, 2019.

[3] Jinyu Wan, Paul Chu, and Yi Jiao. Neural network-based multiobjective optimization algorithm for nonlinear beam dynamics. *Phys. Rev. Accel. Beams*, 23:081601, Aug 2020.

[4] Xiaobiao Huang, Minghao Song, and Zhe Zhang. Multi-objective multi-generation gaussian process optimizer for design optimization, 2019.

[5] Minghao Song, Xiaobiao Huang, Linda Spentzouris, and Zhe Zhang. Storage ring nonlinear dynamics optimization with multi-objective multi-generation gaussian process optimizer. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 976:164273, 2020.

[6] Stephen Hudson, Jeffrey Larson, Stefan M. Wild, David Bindel, and John-Luke Navarro. libEnsemble users manual. Technical Report Revision 0.7.0, Argonne National Laboratory, 2020.

[7] Qianqian Su, Jeffery Larson, Weiming An, Fei Li, Yujian Zhao, Lance Hildebrand, Stefan Wild, and Warren Mori. Optimization of beam qualities on Plasma Wakefield Acceleration. In *APS Division of Plasma Physics Meeting Abstracts*, volume 2019 of *APS Meeting Abstracts*, page GO5.006, January 2019.

[8] xopt: Simulation optimization, based on deap. `https://github.com/ChristopherMayes/xopt`.

[9] Anubhav Jain, Shyue Ping Ong, Wei Chen, Bharat Medasani, Xiaohui Qu, Michael Kocher, Miriam Brafman, Guido Petretto, Gian-Marco Rignanese, Geoffroy Hautier, Daniel Gunter, and Kristin A. Persson. Fireworks: a dynamic workflow system designed for high-throughput applications. *Concurrency and Computation: Practice and Experience*, 27(17):5037–5059, 2015.

[10] Remi Lehe. Using Gaussian Processes to target the injection boundary in laser-plasma simulations, 2019. 2nd ICFA Mini-Workshop on Machine Learning for Charged Particle Accelerators.

[11] Ocelot generic optimizer for accelerators. `https://github.com/ocelot-collab/optimizer`.

[12] Axel Huebl. Develop/integrate data standards start-to-end workflows for Accelerator Physics. *Snowmass21 LOI*, 2020.

[13] Jean-Luc Vay. EVA (End-to-end Virtual Accelerators). *Snowmass21 LOI*, 2020.